
Combining Unity with Google Spreadsheets

User Manual
Version 1.0.5

LITTERATUS


http://www.litteratus.net

Email: googlefu@litteratus.net

# Introduction

### What is GoogleFu?

GoogleFu is a tool that works from within the Unity editor to assist in the creation of database-like objects or files, and provide the means to access this data from within Unity. All of the processing and network access is done prior to building the game, so there is no file access, and no network communication needed once the game is built.

### Why do I need this?

The use of spreadsheets allows you to better visualize data in relation to other entries.
By creating spreadsheets using Google Spreadsheets directly, or by using other popular spreadsheet formats, you are able to design and create a data driven workflow for your game very quickly and easily. GoogleFu provides the mechanism to quickly interpret the spreadsheets, and writes custom classes to either access directly, or through a game object in the scene.

### What are some examples of how to effectively use GoogleFu?

One of the most common uses for Spreadsheets is string localization. Google Spreadsheets provide a formula to assist in translating entries from one language to another. This may not be a perfect translation, but it's enough to get you started. And because Google allows you to collaborate with others seamlessly, providing a localization team the spreadsheet is very easy. Once strings are translated, GoogleFu will be able to import the localized strings into the project, and be ready to access within seconds.

Another use is creating templates for units in your game, loot tables, item stats, enemy stats, asset file mapping, and much more. Imagine creating AI for your game, and being able to visualize and tweak health, speed, and other stats of each unit without editing a single line of code. Or adding all of your sound hooks in the game by ID, and being able to change which sound is being played by simply editing an entry in a spreadsheet.

### Where does the data come from?

You provide all of the data needed. There are a few rules to follow when creating a spreadsheet, but it is up to you to provide any type of data you wish. Whether your spreadsheet contains a single column of information, or hundreds, GoogleFu will be able to handle it.

## What spreadsheet formats does GoogleFu support?

GoogleFu will only access Google Spreadsheets directly, however Google Spreadsheets is able to import many different types of files, including: .xls, .xlsx, .ods, .csv, .txt, .tsv, .tab

## Am I required to use a Google account to use GoogleFu?

Short answer: No.
Long answer: It's recommended, yes. The majority of GoogleFu's functionality revolves around accessing the account of a user, and allowing the user to select individual pages of a workbook to import into the Unity project. While you are able to manually add public, published workbooks and pull the data from them, this would mean that anyone who knew where to find your workbooks could access them. By providing GoogleFu with your account information, it is able to access your private workbooks.

## Isn't using my Google account a security risk?

Only if you choose to save your credentials in the GoogleFu editor. All of the source code for GoogleFu is exposed. We chose to do this as a verification of trust. GoogleFu itself isn't doing anything with your account information other than providing it to the Google API where necessary. However if you choose to save your credentials, it is saved in the registry, as plain text, so that you do not have to provide them the next time you open the window. By manually logging out, these values are cleared, and GoogleFu will not save them. We attempted to get OAuth2 working with GoogleFu so that your information was guaranteed to be safe, but the current Monodevelop restrictions made it impossible.

## Who is Litteratus?

Litteratus is a small independent studio consisting of two industry veterans. We have been professionally employed in the Video Games / Military Simulation industries for over 8 years, and Litteratus is our hobby. Together we have worked for companies and have credits on over 15 different games for PC, Nintendo DS, and PSP, and are currently working on an unannounced AAA Xbox 360 title. GoogleFu is a product of our hobby work, and a tool that we developed to assist in our own creations. We decided to productize it as we believe it would be useful to anyone, not just Litteratus itself.

# Getting Started

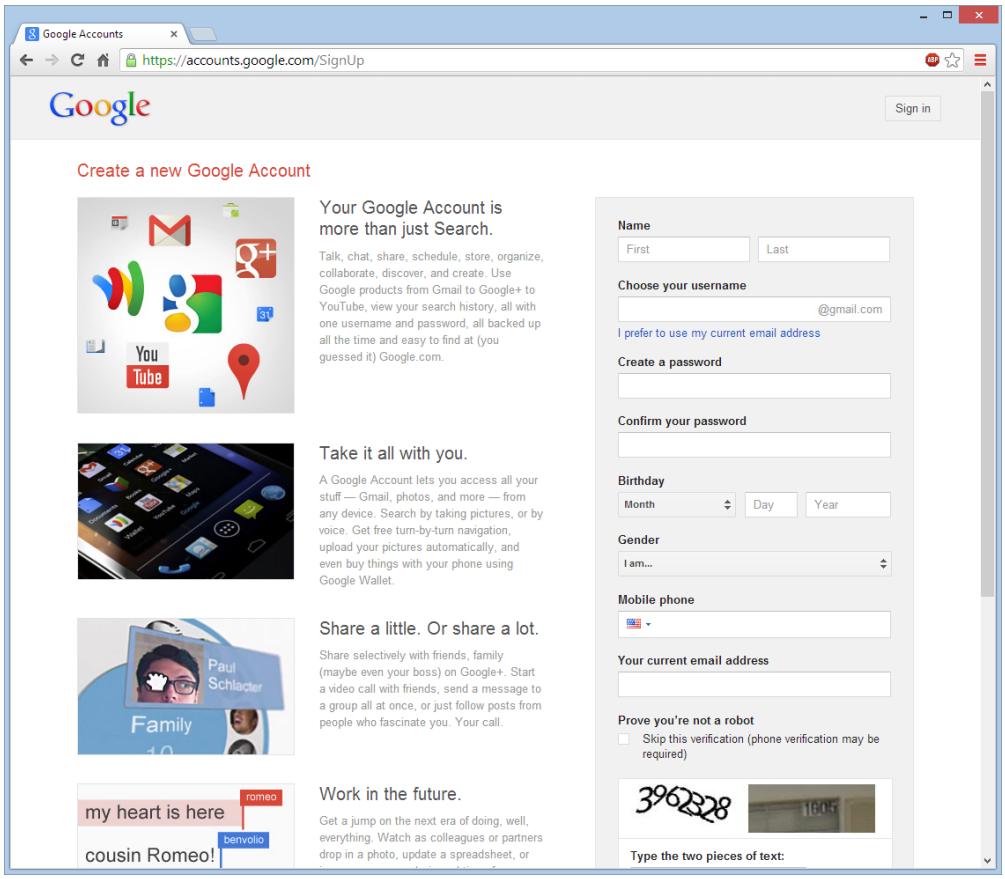
## Setting up GoogleFu for your project

### Signing up for a Google account

Whether you decide to provide GoogleFu with your authentication or not, this step is mandatory. You must have a Google account in order to use GoogleFu. If you already have a Google account, and wish to use this account with GoogleFu, you may skip to the next step.

You can create a new Google account at the following website:
[Google Account Creation]

Once you have navigated there using the web browser of your choice, you will need to create an account. Fill in the following form with your account information.

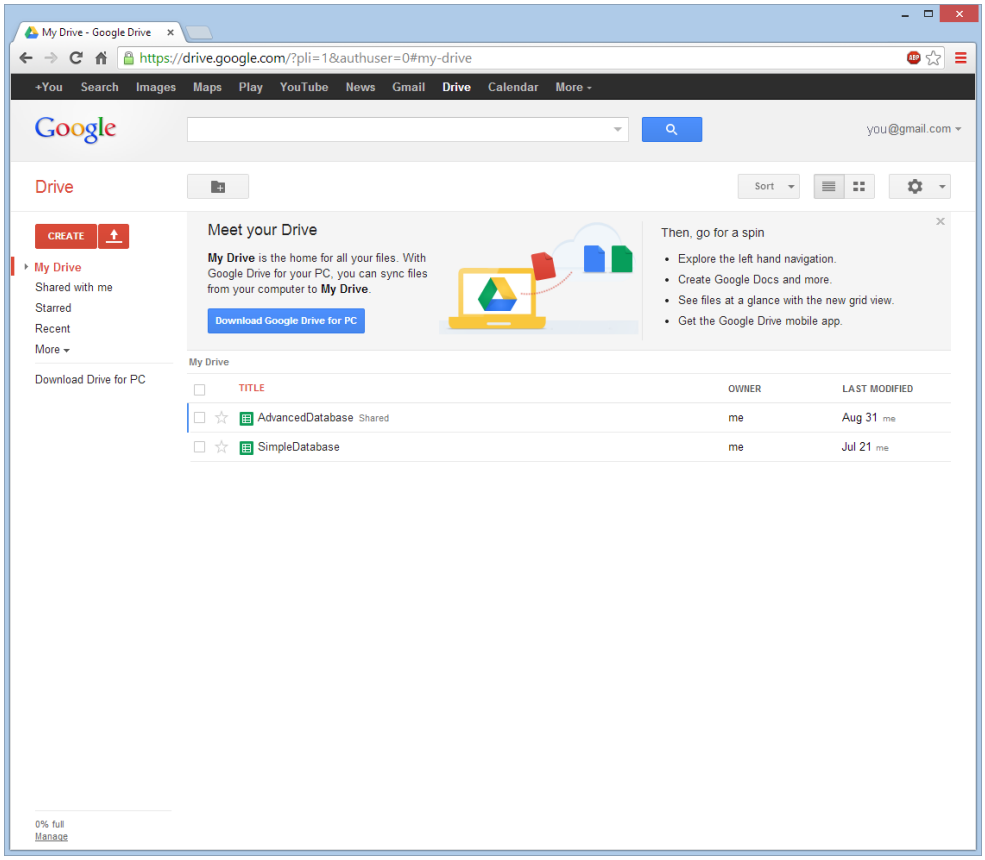


## Accessing your Google Drive

Once you have created your new account, you will be able to sign into your Google Drive. This is where all of your Google Spreadsheets will be located. The online access is provided free of charge and you are given a few gigabytes worth of space to save all of your work. It is cloud based and can be accessed anywhere, from any computer with access to Google, provided that you authenticate with Google and log in to your Google Drive.

Your Google Drive can be accessed from here:
[Your Google Drive]

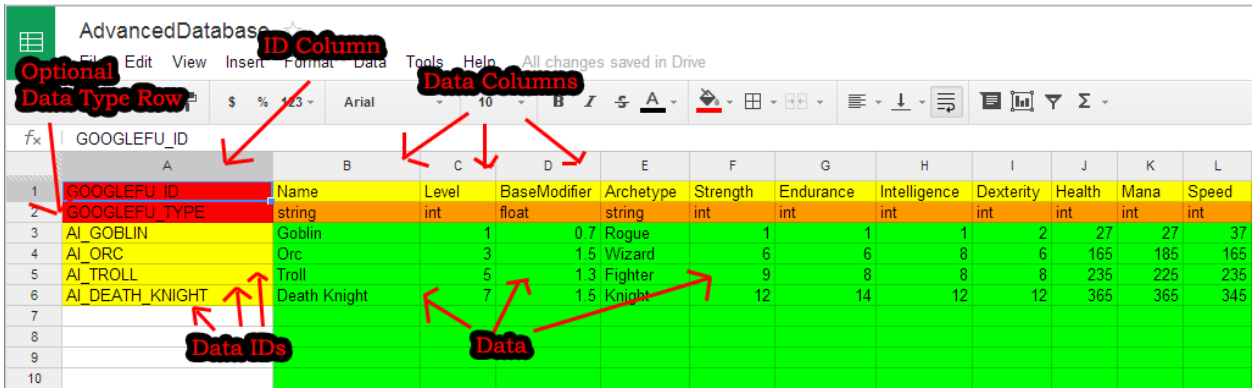
The landing page for your Google Drive should look similar to this:




## Setting up a Spreadsheet so GoogleFu can parse it

Spreadsheets have 2 basic column types, ID Columns and Data Columns. The values in ID Columns will be the way that GoogleFu indexes the data for retrieval. The values in Data Columns is what will be retrieved when accessing the database. In addition to columns, there are 3 Row types, Header, Data Type, and Data rows. The first row in the spreadsheet will always be the Header row, this contains the variable names of your data. The second row is the

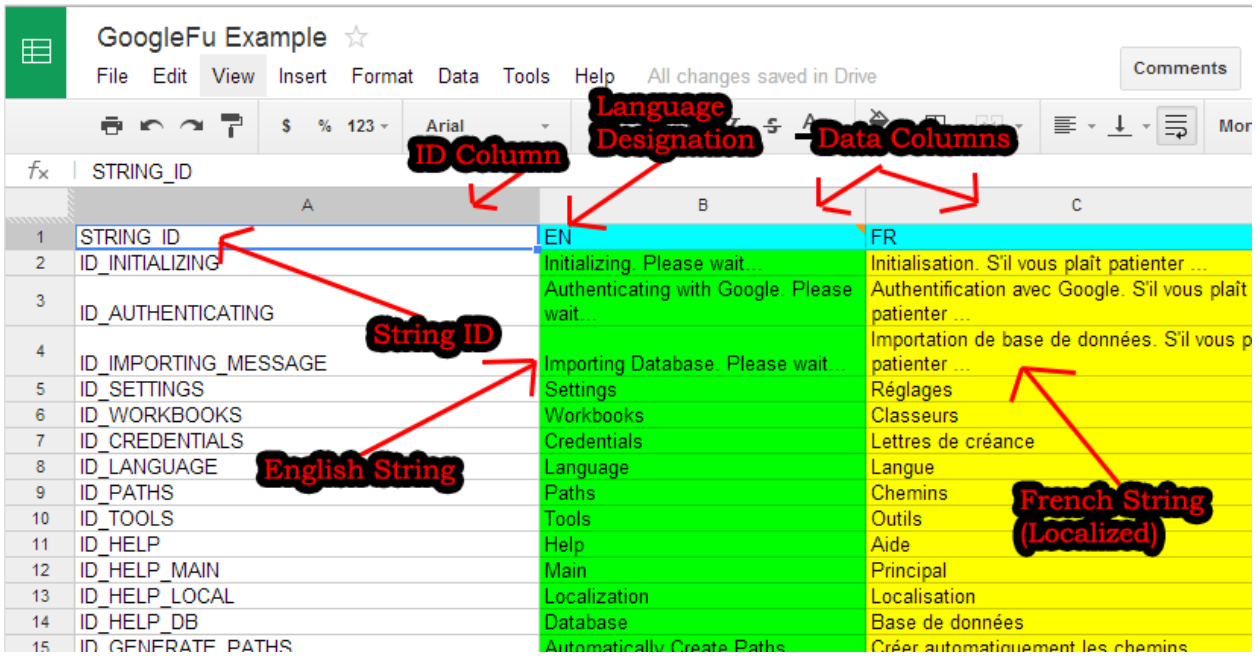
Data Type row, and is completely optional. If your spreadsheet contains a Data Type row, the final data classes will contain strongly typed values of the types specified, otherwise they will be simple strings and it will be up to you to convert them if necessary. The remaining rows are the Data rows, and contain all of your data entries into the spreadsheet.




| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | GOOGLEFU_ID | Name | Level | BaseModifier | Archetype | Strength | Endurance | Intelligence | Dexterity | Health | Mana | Speed |
| 2 | GOOGLEFU_TYPE | string | int | float | string | int | int | int | int | int | int | int |
| 3 | AI_GOBLIN | Goblin | 1 | 0.7 | Rogue | 1 | 1 | 1 | 2 | 27 | 27 | 37 |
| 4 | AI_ORC | Orc | 3 | 1.5 | Wizard | 6 | 6 | 8 | 6 | 165 | 185 | 165 |
| 5 | AI_TROLL | Troll | 5 | 1.3 | Fighter | 9 | 8 | 8 | 8 | 235 | 225 | 235 |
| 6 | AI_DEATH_KNIGHT | Death Knight | 7 | 1.5 | Knight | 12 | 14 | 12 | 12 | 365 | 365 | 345 |
| 7 | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | |

If you are setting up a Localization spreadsheet, you will not need to use the Data Type row, as all of the data is going to be strings anyway. Google has a handy formula for automatically translating cells:
=GoogleTranslate( value, from language id, to language id)

So, using the following example:



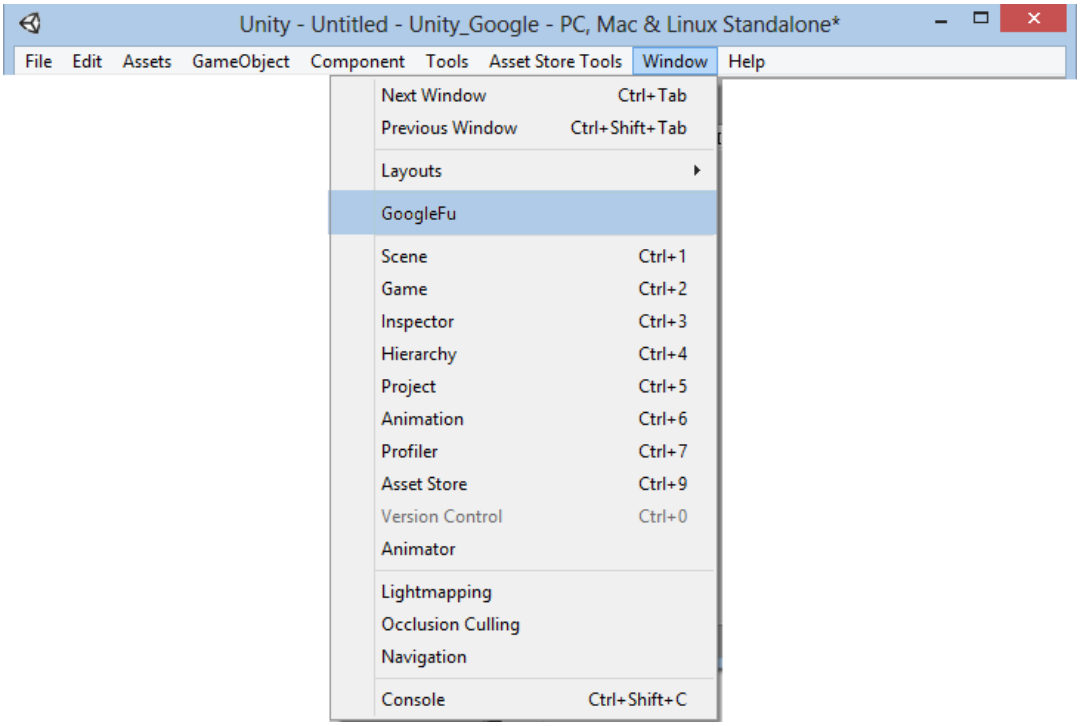

| | A | B | C |
|---|---|---|---|
| 1 | STRING_ID | EN | FR |
| 2 | ID_INITIALIZING | Initializing. Please wait... | Initialisation. S'il vous plaît patienter ... |
| 3 | ID_AUTHENTICATING | Authenticating with Google. Please wait... | Authentification avec Google. S'il vous plaît patienter ... |
| 4 | ID_IMPORTING_MESSAGE | Importing Database. Please wait... | Importation de base de données. S'il vous p patienter ... |
| 5 | ID_SETTINGS | Settings | Réglages |
| 6 | ID_WORKBOOKS | Workbooks | Classeurs |
| 7 | ID_CREDENTIALS | Credentials | Lettres de créance |
| 8 | ID_LANGUAGE | Language | Langue |
| 9 | ID_PATHS | Paths | Chemins |
| 10 | ID_TOOLS | Tools | Outils |
| 11 | ID_HELP | Help | Aide |
| 12 | ID_HELP_MAIN | Main | Principal |
| 13 | ID_HELP_LOCAL | Localization | Localisation |
| 14 | ID_HELP_DB | Database | Base de données |
| 15 | ID_GENERATE_PATHS | Automatically Create Paths | Créer automatiquement les chemins |

If you want to translate the cell B2 from English to French, you would use the following in Cell C2:
=GoogleTranslate($B2,$B$1,C$1)

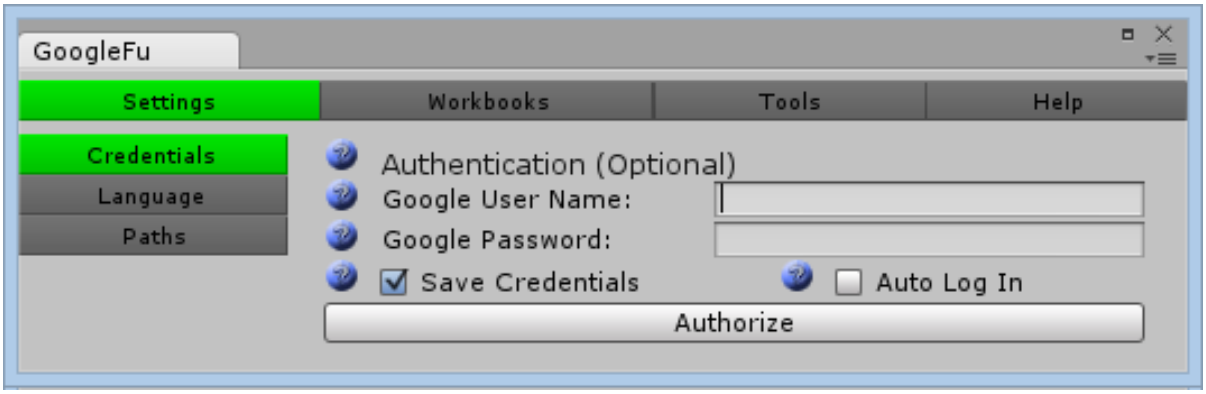
Google Spreadsheet uses the Google Translate functionality internally, and is not a feature of GoogleFu. If you have a large enough string database, and are localizing for many languages, you will find that sometimes Google will stop translating. This is because it only allocates so much time to do translations. A way to resolve this is wait for the sheet to fully load in Google and search for “#N/A” on the page. It will show you cells that have not been fully resolved. If you add or remove a space in the English cell, it will attempt to resolve only the dependent cells. Repeat this process for any unresolved cells.

## Opening GoogleFu

When you start GoogleFu for the first time by opening the Unity Editor and selecting Window->GoogleFu:



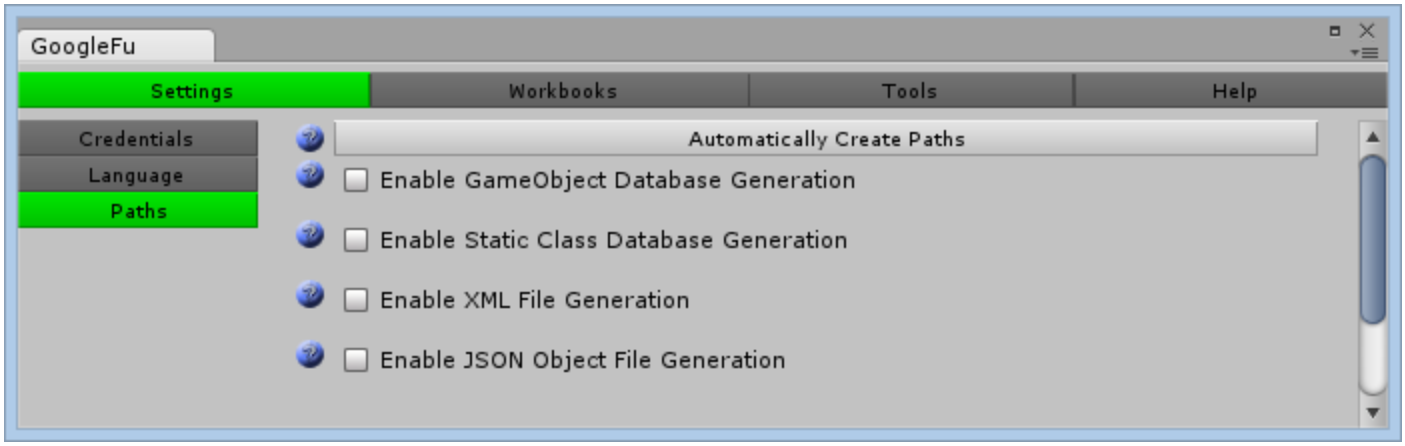

You will be presented with the GoogleFu window:




At the top there are four tabs, Settings, Workbooks, Tools, and Help. The Settings tab will be highlighted, and you will have 3 options on the left side, Credentials, Language, and Paths.

## Setting up Paths in GoogleFu

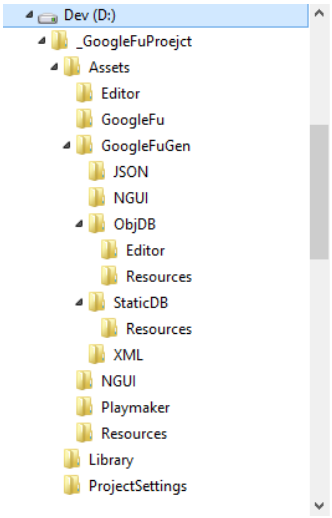
Click the Paths button and you will see the following screen:




Each of the options allows you to enable exporting of a specific type of database. But before GoogleFu will allow you to export anything, you must select appropriate folders within your project path to export the generated files. Every time you export, file names are generated based on the contents of the spreadsheet, you will be overwriting the existing files. Choose directories that are safe to automatically generate files in.

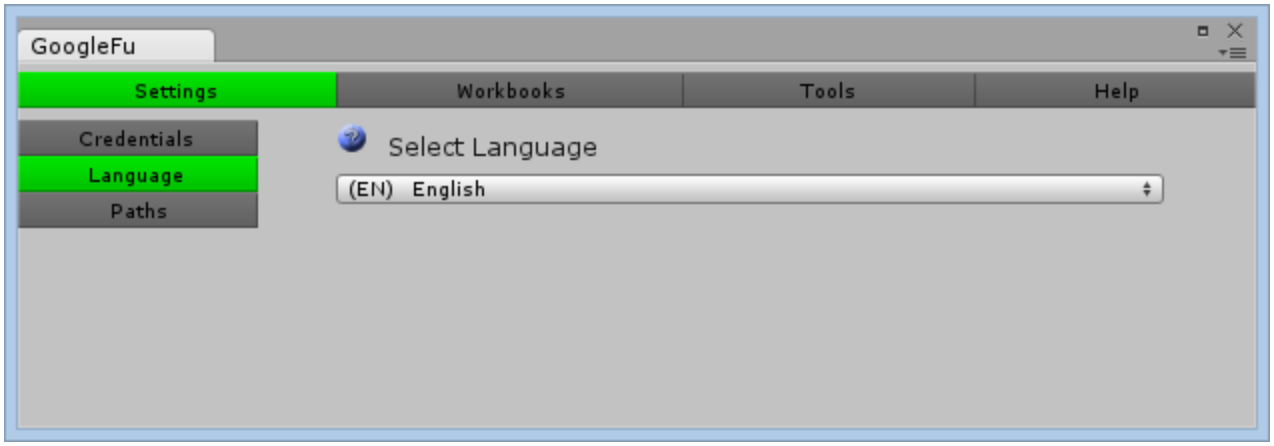
| |
|---|
| GoogleFu will warn you if a path you have chosen is invalid, and as a precaution require you fix it before it will function. |

The “Automatically Create Paths” button will set up all of the paths for you. A Directory called “GoogleFuGen” will be created, and all generated files will be placed in subdirectories of this folder.




## Setting the GoogleFu language (Optional)

Click the Language button and you will see the following screen:



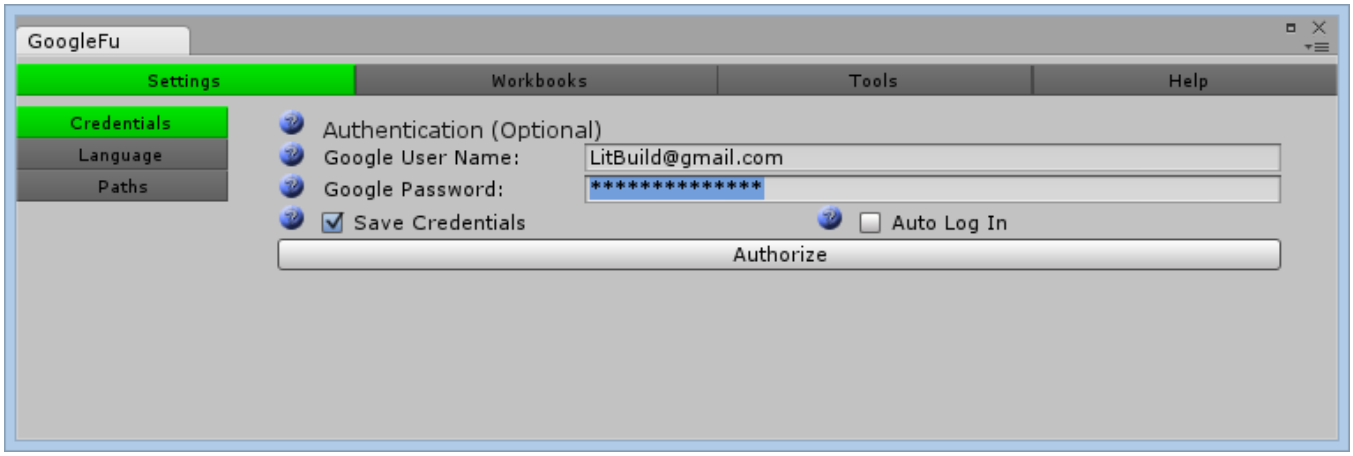

GoogleFu has utilized a Google Spreadsheet and Google Translation  to localize the plugin itself. You may choose a language to use for GoogleFu by choosing an option in the drop-down box. Google Translation is not a perfect localization for every string, and the result is only an initial rough translation.

| |
|---|
| Workbook names and Sheet names are pulled directly from Google, and will remain in the language that is in the Google Spreadsheet. |

## Authenticating with Google and selecting an Active Workbook

### Logging into your Google account with GoogleFu (Optional)

In the Credentials page under the Settings tab you will find a standard login dialog. Here you will input your Google user name and password. This step is optional because you are also able to manually enter published workbook url's as well. The reason you would want to use your Google account is to allow GoogleFu to access workbooks that are privately associated with that account.




The "Save Credentials" option will let GoogleFu store your credentials so you do not have to manually enter them each time you open the GoogleFu window. Note that this is a security risk. The way GoogleFu stores your credentials is in plain text, in the registry. This means the username and password is available to anyone who knows where to look.

The "Auto Log In" option is only available if "Save Credentials" is checked. This will allow GoogleFu to automatically authenticate with Google every time the window is opened. Note that authentication is done synchronously, and may stall the loading process until GoogleFu has successfully authenticated. If you are experiencing what seems like a freeze when opening a project with the GoogleFu window visible, then you may want to uncheck this option, and authorize with Google only when necessary.

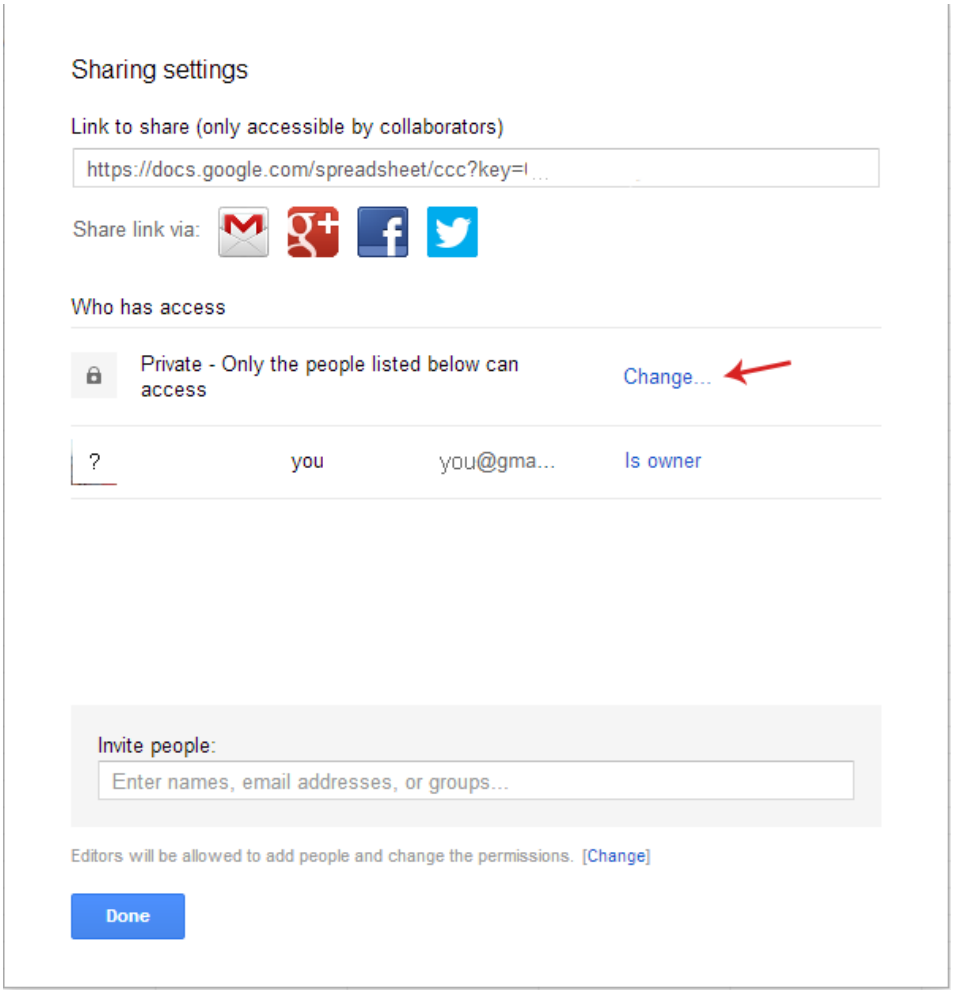
Once you are logged in, the screen will show the currently logged in account name, and a "Log Out" button. If "Save Credentials" is unchecked when you log out, this will clear all account data stored, and you will have to re-input your username and password to access your account again.

## Using Manual Workbooks (Optional)

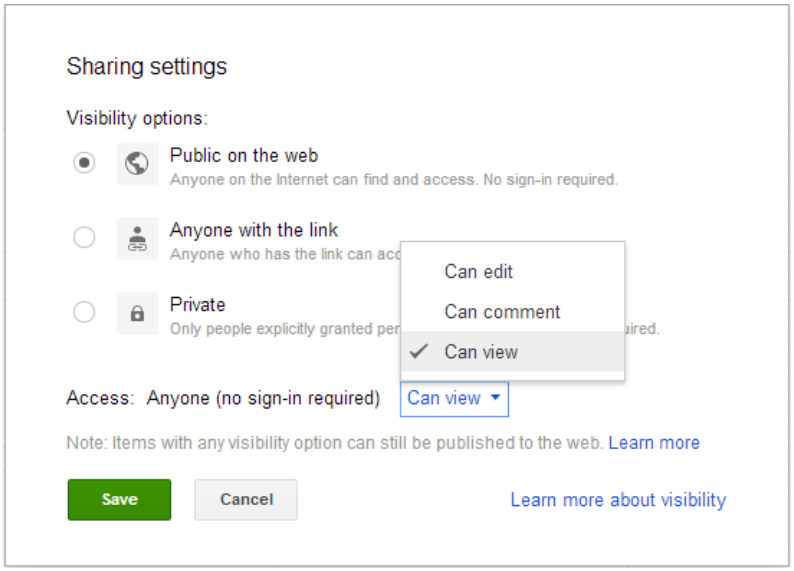
If you are planning to use a published workbook, you will need to tell GoogleFu how to access this workbook. You will need to first have the URL of the published workbook, to let GoogleFu know where it is located. If you have authenticated with your Google credentials, GoogleFu will have access to your workbooks, and this step is purely optional.

The Workbook you are attempting to access must be public, and published to the web. If you are attempting to access a workbook that is private, you will need to authenticate with Google to view it.

To set the workbook to public, open it in Google, open the File menu and select “Share...”. In the Sharing Settings dialog you will need to change the access type by clicking the “Change…” link.

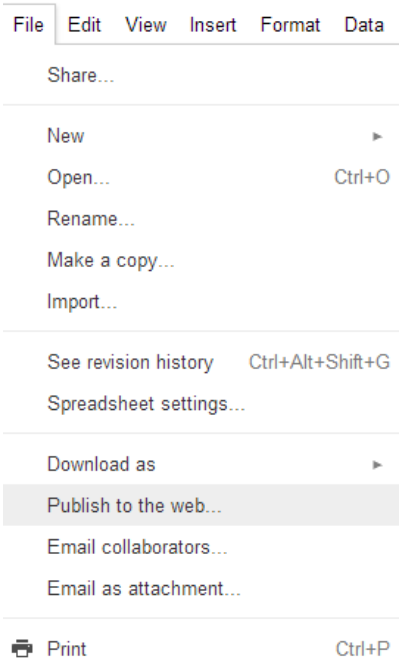


Another dialog will appear allowing you to select the sharing settings. Choose “Public on the web” option. If you wish to allow anyone to edit the workbook publically you may select the “Can edit” option, or leave it as “Can view” if you simply wish to allow GoogleFu to access the data.

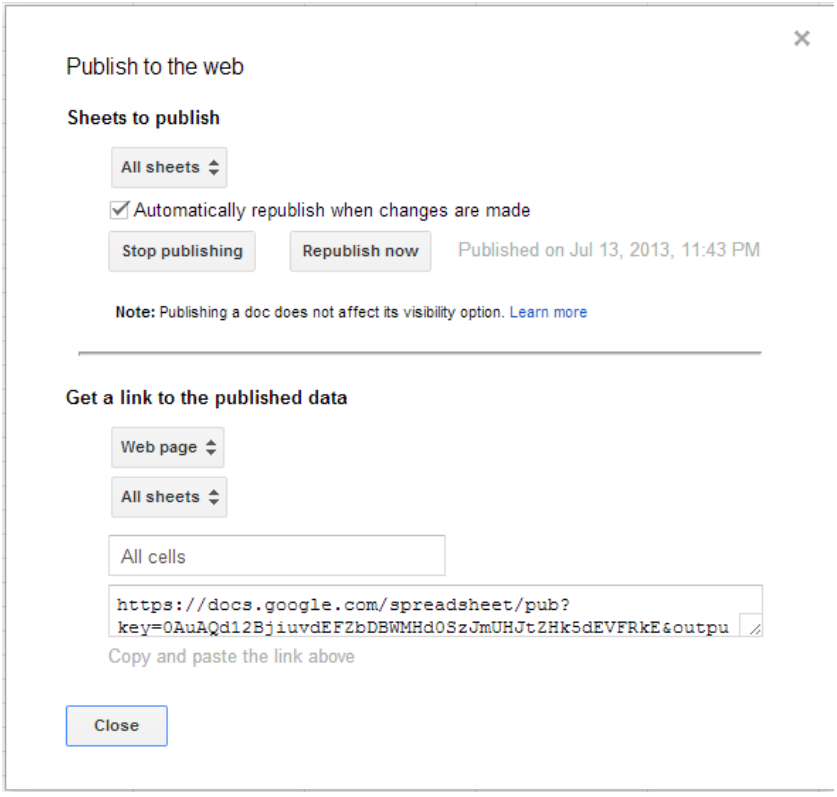



Save, and exit the dialog. Next you will need to publish the workbook.

To find the published workbook URL, open the spreadsheet in Google and open the File menu, and select “Publish to the web…”

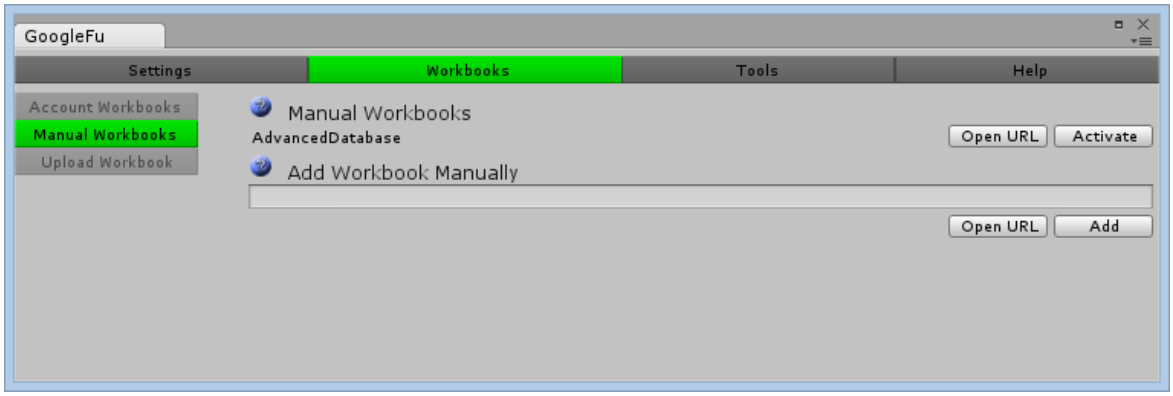


You will then see a dialog that gives you options for publishing the worksheet. Leaving all of the options as defaults and choosing the “Publish Now” option, Google will generate a URL for the workbook.



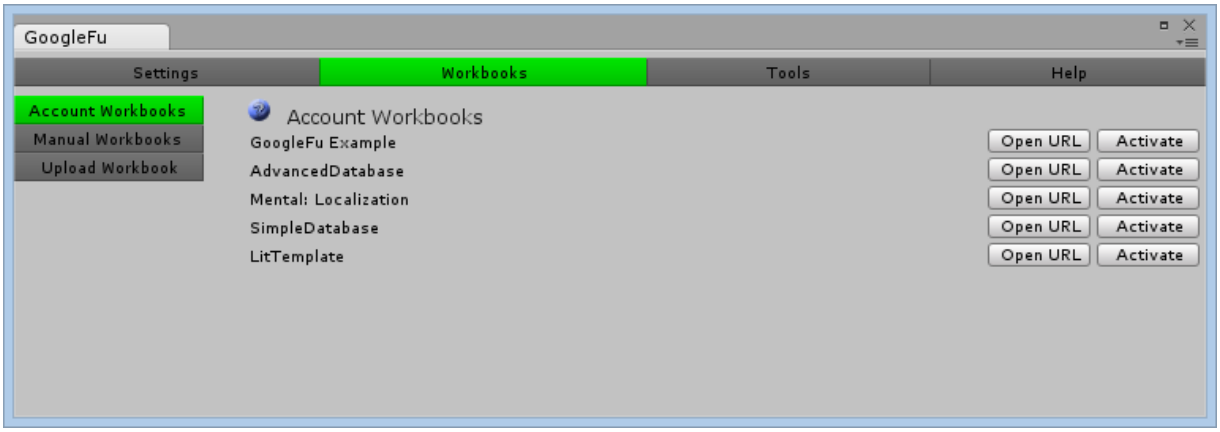

The URL in the bottom box is the URL that you will be giving GoogleFu. Copy it from the box.

Once you have published the workbook, return to the GoogleFu window and navigate to the Manual Workbooks page of the Workbooks tab. There you will paste the URL obtained from Google into the box, and click the “Add” button. The workbook will appear under the “Manual Workbooks” heading.



## Using Account Workbooks (Optional)

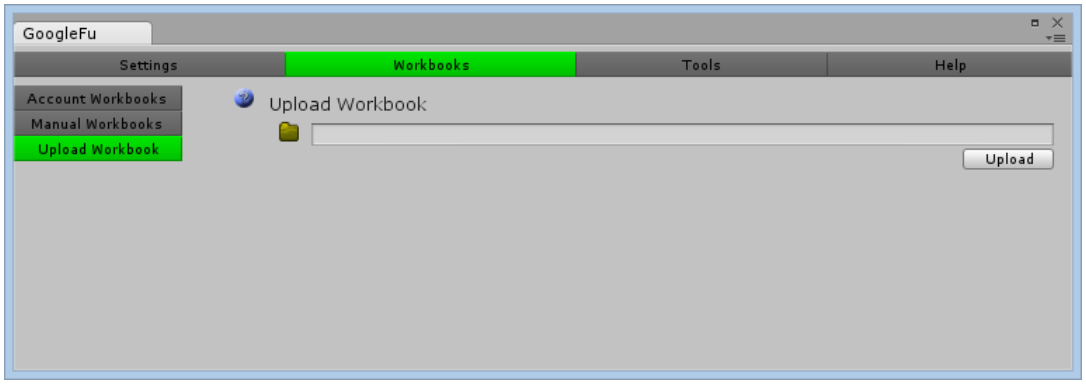
If you have authenticated with your Google Credentials, the workbooks associated with your account will be available in the “Account Workbooks” page of the “Workbooks” tab.




To quickly open the workbooks for editing, you can use the “Open URL” button, and it will open in your web browser.

## Uploading Workbooks (Optional)

If you have a spreadsheet on your drive, GoogleFu can assist in uploading that to Google Spreadsheets by using the “Upload Workbook” page in the Workbooks tab.

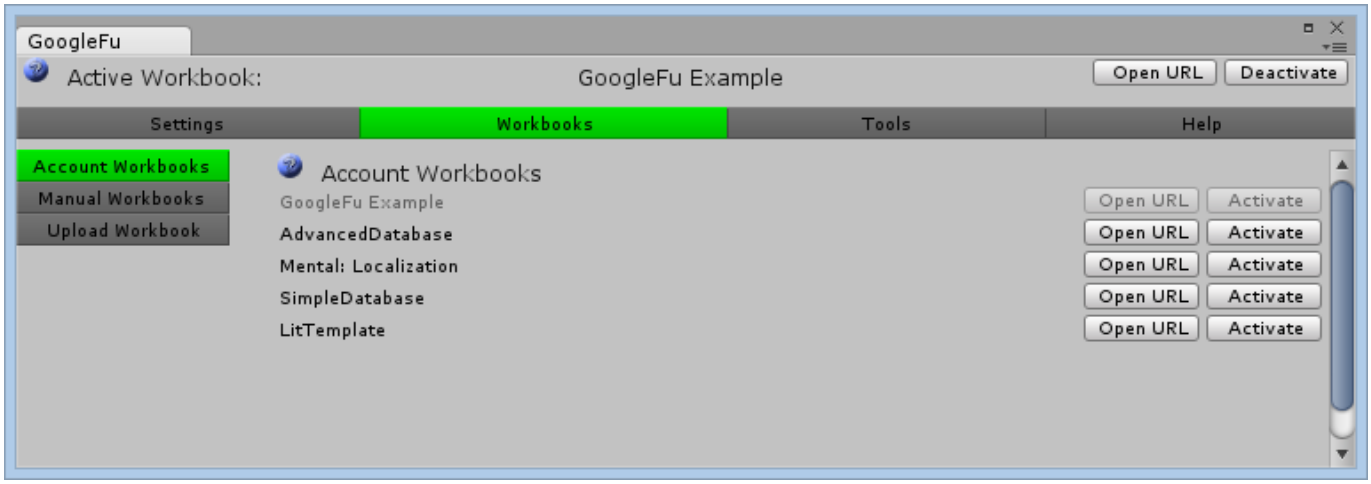



Google is able to process the following file types: .xls, .xlsx, .ods, .csv, .txt, and .tsv
Use the folder icon to browse to a Spreadsheet and click the “Upload” button.

| Uploaded spreadsheets will use “Untitled” as the name until you change it manually in Google |
|---|

### Activating a Workbook

To export data from a workbook you will first need to activate the workbook you plan to export from. In either the “Account Workbooks” or “Manual Workbooks” page under the “Workbooks” tab, you can activate a workbook by clicking the “Activate” button next to the entry.



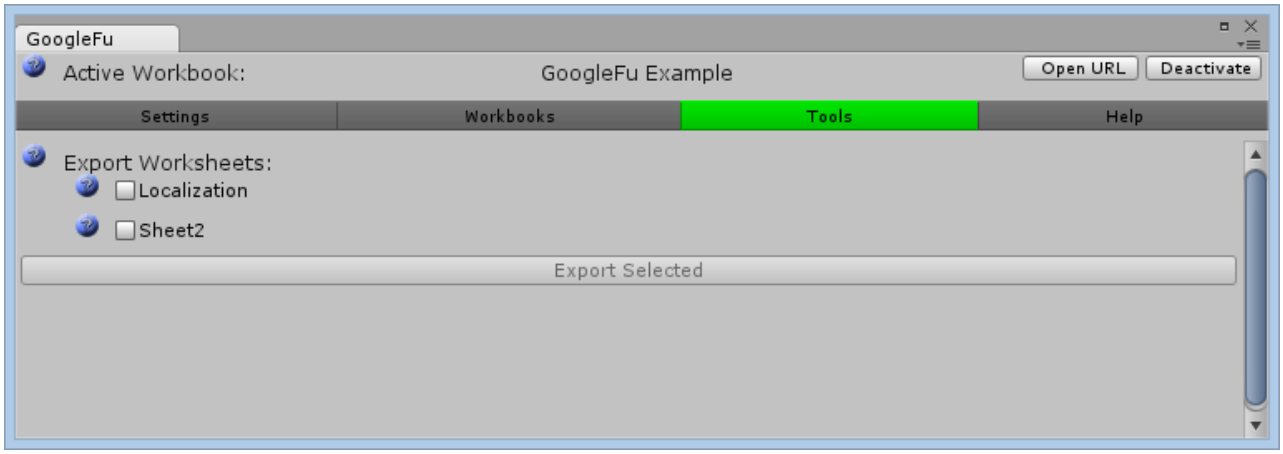

When a workbook is activated it will turn grey in the list, and appear above the main tabs.
To deactivate the active workbook you may either click the “Deactivate” button in the active workbook above the tabs, or simply activate a different workbook.

## Exporting data from a Google Spreadsheet

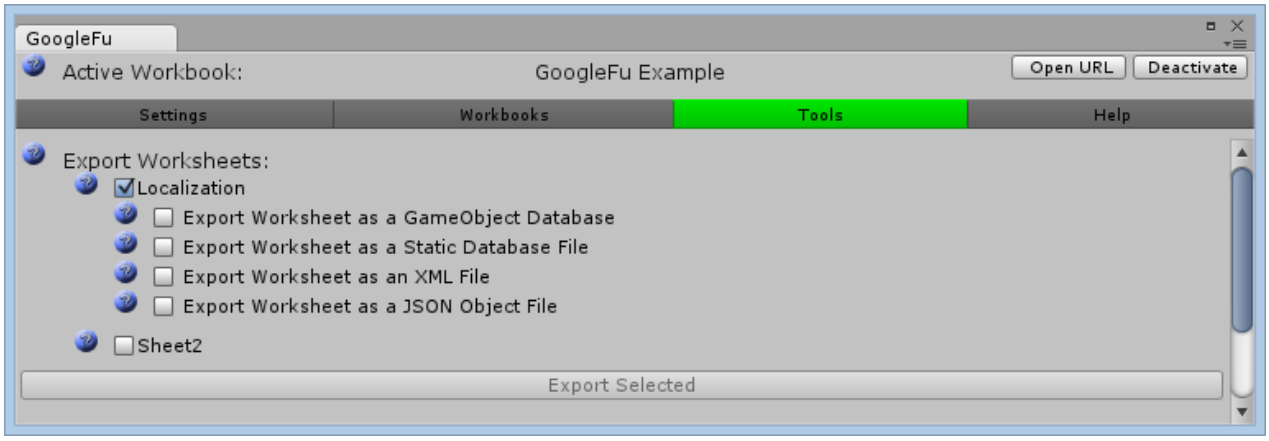
Once you have an active workbook, and the export options are set up in the “Paths” page under “Settings”, the “Tools” tab will become available. The Tools tab is where you will be able to export the data from a Google Spreadsheet, and optionally import it into Unity as a Object based or Static database.

### Selecting Worksheets to export

Select the Tools tab once you have activated a workbook, and you will see a page that lists all of the available worksheets contained in the workbook. Each sheet is exported individually. You are able to select which worksheets are going to be exported, and which format the data will be in once it is exported.




Click the checkbox next to each sheet you wish to export, and you will be presented with options for the format types available to you.




## Choosing an export type

### GameObject Database

A GameObject that contains a generated script will either be created for you, or you may specify a GameObject to attach the script to. The data from the worksheet will be inserted into the GameObject via this generated script, and be retrieved using functions generated in the script.

-Option: "Use first row as Value Types"

The first row of the worksheet can be used to specify value types ("string", "int" etc.). If this option is used, the data structure that is generated will be strongly typed based on the values in this row. This saves having to convert data from strings at runtime.

-Option: "Use the DoNotDestroy method on the generated GameObject"

If you initialize a database, and wish for the database to remain loaded indefinitely, using this option will ensure the GameObject is created once, and will not be destroyed between scenes.

-Option: "Select an existing GameObject to attach the Database to"

You may choose a GameObject from the scene to attach the generated script to, or leave it as None to automatically create a “databaseObj” GameObject.

-Option: “Generate Playmaker Actions”

If you have Playmaker installed, this option will be available to you. Automatically generate Playmaker actions that assist in accessing data from your Database.

### Static Database

A file will be generated that contains a singleton that you may access from other scripts. All of the data from the worksheet will be inserted into a generated data structure on creation of this singleton class.

-Option: “Use first row as Value Types”

The first row of the worksheet can be used to specify value types (“string”, “int” etc.). If this option is used, the data structure that is generated will be strongly typed based on the values in this row. This saves having to convert data from strings at runtime.

### XML File

An XML file is generated with the data from the worksheet.

### JSON Object File

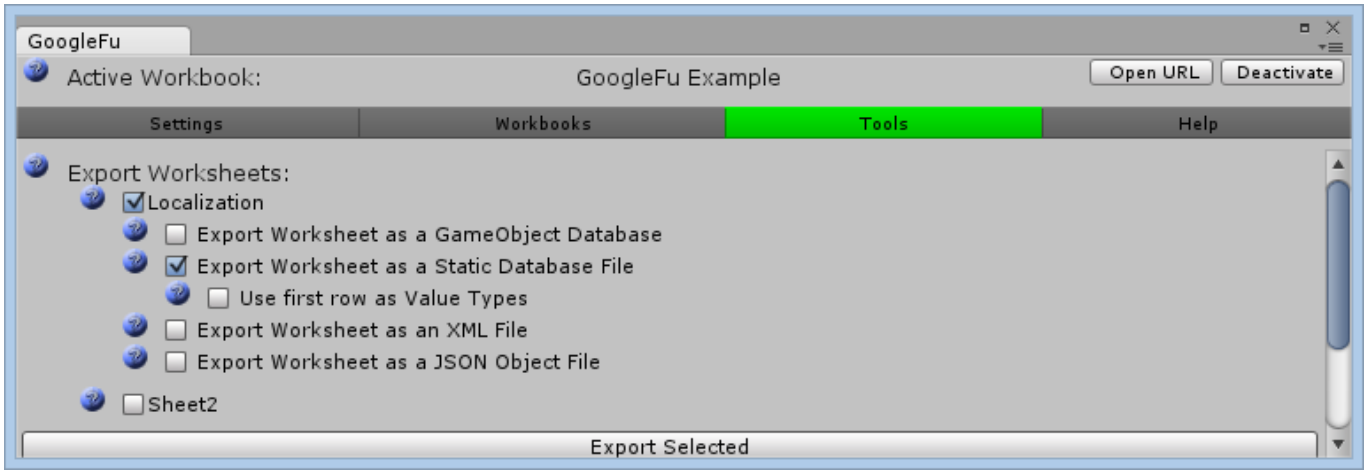
A file containing the code for a JSON object is generated with the data from the worksheet.

### NGUI Localization

If NGUI is installed in the project this option is available. This will export data from the worksheet in a format that NGUI uses for localization. Note: It is up to the user to ensure the column headers are in a format that NGUI recognizes.

### Daikon Forge Localization

If Daikon Forge is installed in the project this option is available. This will export data from the worksheet in a format that Daikon Forge uses for localization. Note: It is up to the user to ensure the column headers are in a format that Daikon Forge recognizes.



Once you have chosen an export type, and any options that export type has available, you can click the “Export Selected” button to generate the files and/or GameObjects that contain the data from your worksheet. The generated files will be written to the directories you have set up in the “Paths” page under “Settings”.

# Accessing your data

After exporting the worksheets, you will want to access your data. Depending on how you exported, this is done in different ways. The XML and JSON files are designed to be used with other tools, or read into your game manually. The NGUI Localization files are used by NGUI directly, and the NGUI Localization documents will illustrate how to use it. The GameObject Database and Static Database have generated functions that allow you to access the data. Below are examples on how to use them.

## GameObject Database Data Retrieval

A GameObject Database is an object that exists in the scene, and has all of the database information stored within it. It contains a script that is generated by GoogleFu that provides access to the data contained in it. You can access the data by first gaining access to the script.

The code below is a snippet from the GoogleFu Sample

```
// Example of an Object based Database.
CharacterStats _statsDb;

// Since our database exists within an object in the scene, we'll need to find it.
// Alternatively you could expose the database as a public member, and use the
// inspector to set it.
GameObject statsdbobj = GameObject.Find("databaseObj");
if ( statsdbobj != null )
{
        // Get the CharacterStats component out of the GameObject.
        // CharacterStats is the name of the worksheet in the google spreadsheet
        //  that we are getting the monster information from
        _statsDb = statsdbobj.GetComponent<CharacterStats>();
}
```

Once you have access to the script, accessing the data is simple

To access entire rows, use the code below

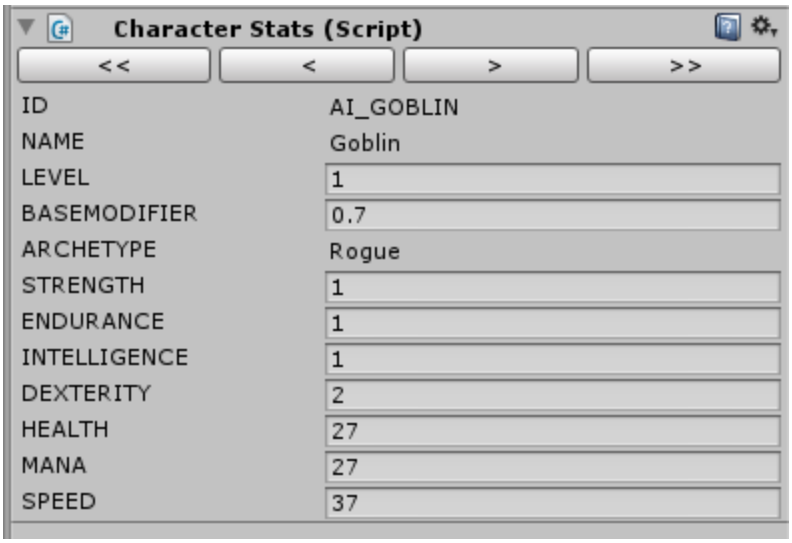
```
// The rowIds enumeration is generated from the first column of the spreadsheet
// It serves as a lookup table for accessing specific data rows
_statsDb.Rows[ ( int )CharacterStats.rowIds.AI_TROLL ];

// You can also access data randomly
_statsDb.Rows[ Random.Range( 0, _statsDb.Rows.Count ) ];
```

```
// You may also use the GetRow function
// GetRow accepts either the rowIds enum
_statsDb.GetRow( CharacterStats.rowIds.AI_GOBLIN );

// GetRow also accepts a string
_statsDb.GetRow( “AI_GOBLIN” );
```

Object Databases can be used in conjunction with Playmaker, and also have custom inspector that is generated to view the data.



## Static Database Data Retrieval

A static database is not tied to a GameObject, but rather a static class that can be accessed anywhere. The data is stored in a variable inside of the class, and is populated when the class is instantiated. Static databases should be instantiated at load time, not during gameplay, as it does take time to populate the database.

The code below is a snippet from the GoogleFu Sample

```
// Example of a Static Database.
Items _itemsDb;

// The Items database is a Static class. Use it by grabbing the .Instance, this will
// ensure the database is correctly initialized. Larger databases may take a while
// to initialize, so grabbing an instance before the game is updating is recommended.
_itemsDb = Items.Instance;
```

To access entire rows, use the code below

```
// The rowIds enumeration is generated from the first column of the spreadsheet
// It serves as a lookup table for accessing specific data rows
itemsDb.Rows[ (int)Items.rowIds.ITEM_RING ];

// You can also access data randomly
_itemsDb.Rows[ Random.Range(0, _itemsDb.Rows.Count) ];

// You may also use the GetRow function
// GetRow accepts either the rowIds enum
_itemsDb.GetRow( Items.rowIds.ITEM_PANTS );

// GetRow also accepts a string
_itemsDb.GetRow( “ITEM_PANTS” );
```

## Retrieving Column Data from a Row

The generated scripts contain variables that are named from the column headers in the spreadsheet. The names have been altered to ensure proper variable names, but are very similar. You will want to open the generated script to verify the column names. To access the data, you have a choice of accessing it directly, or by using the GetStringData function, which will return a string value (Regardless if the variables are strongly typed or not).

```
// Accessing a specific value from a row can be done multiple ways
ItemsRow myRing = itemsDb.Rows[ (int)Items.rowIds.ITEM_RING ];

string RingName1 = myRing ._NAME;

string RingName2 = myRing.GetStringData("name");
```

## Samples

Included with GoogleFu is an example spreadsheet that you can upload to your Google account. This example has 3 worksheets to illustrate the syntax you should use when creating your own GoogleFu compatible spreadsheets. The example is located at:

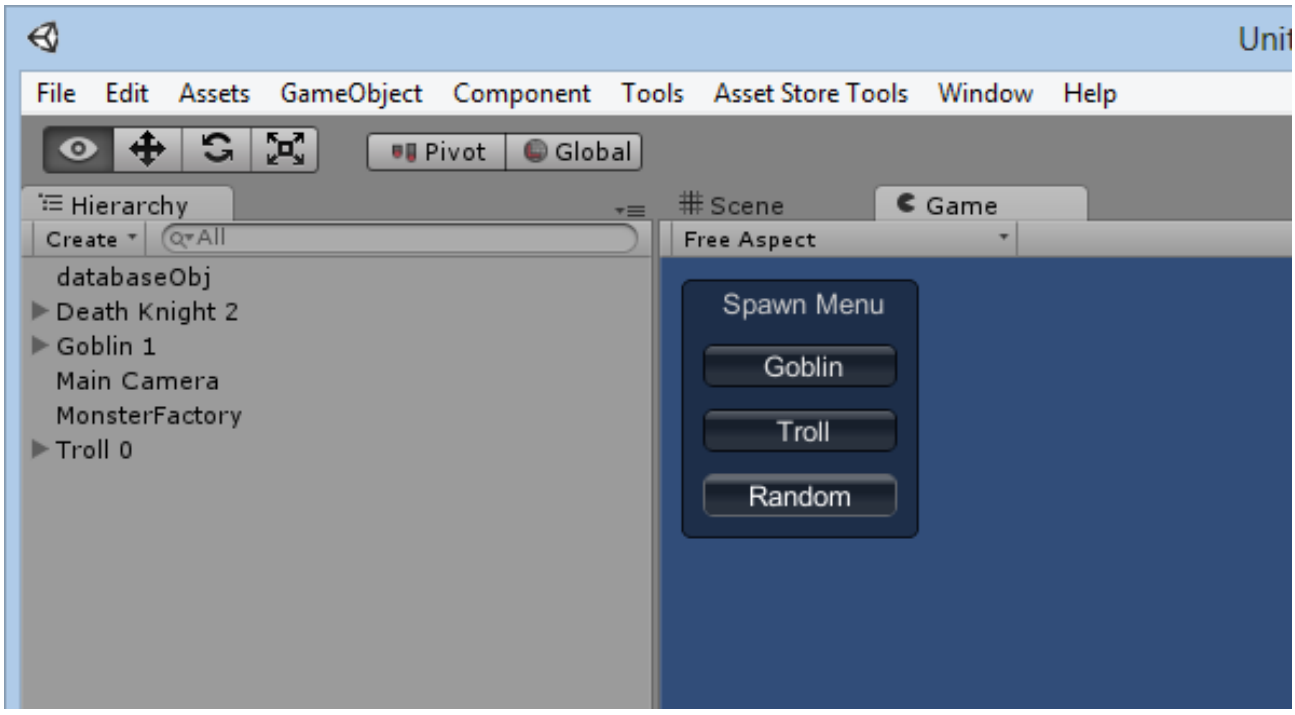
../Assets/GoogleFu/Samples/Data/SampleDatabase.xlsx

> You do not need to be able to open this file locally in order to use it. Simply use the “Upload Workbook” page in the “Workbooks” tab to upload it to Google, and let Google take care of opening it.

Also you will find a sample that uses both Object Database and Static Database to generate dynamic game objects. This example is located at:

../Assets/GoogleFu/Samples/Scenes/DemoScene

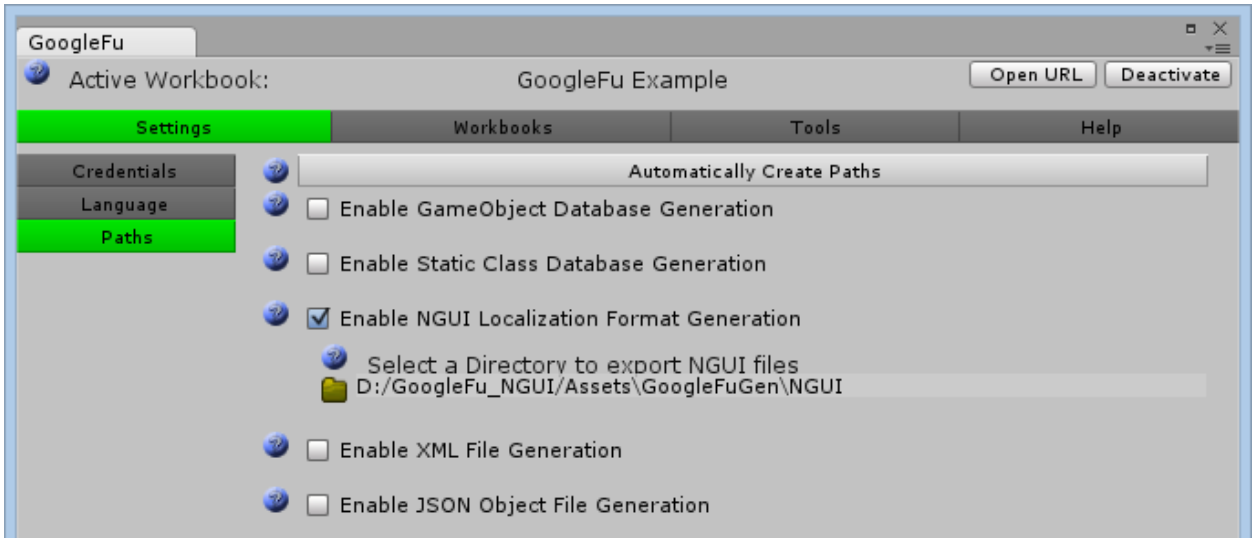
This is a bare bones example of how to use the databases once exported. Simply load the scene and click the buttons to generate a Game Object with data from the database.



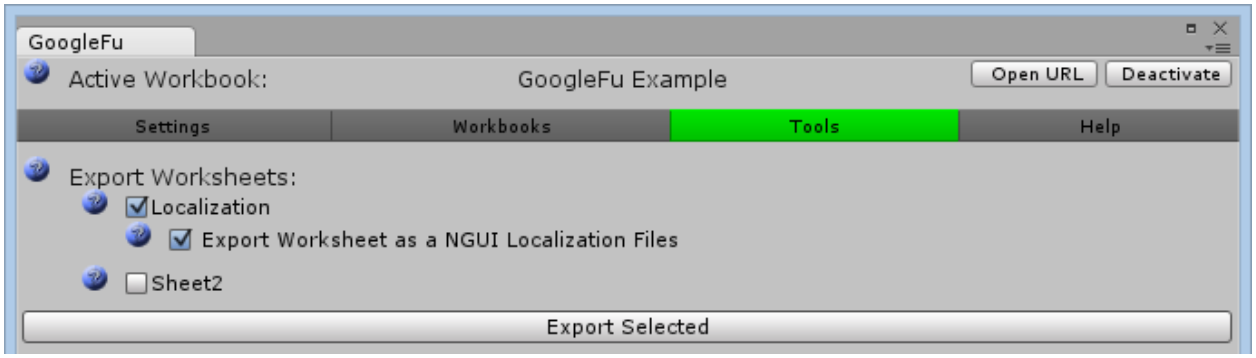
## Using the exported data as NGUI Localization files

If NGUI is present in the project, GoogleFu will recognize it* and allow you to export the database as NGUI Localization files.

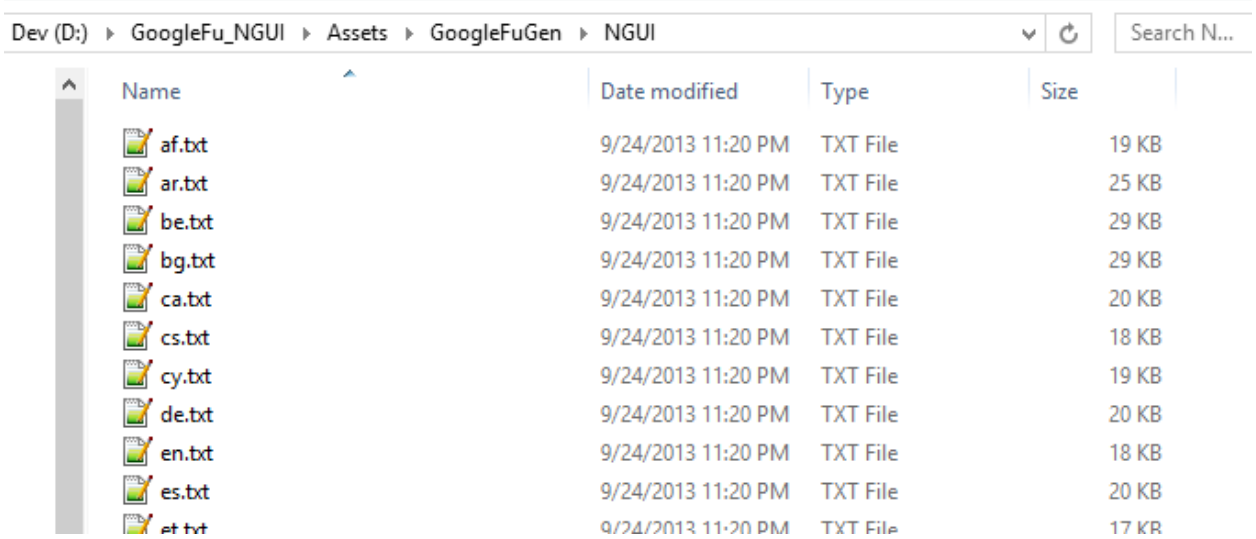
*GoogleFu looks for a specific file that NGUI provides.




Once you have selected an Active Workbook, you will be given the option to export as NGUI Localization Files.



When you export these files, they will be created in the path you have specified.

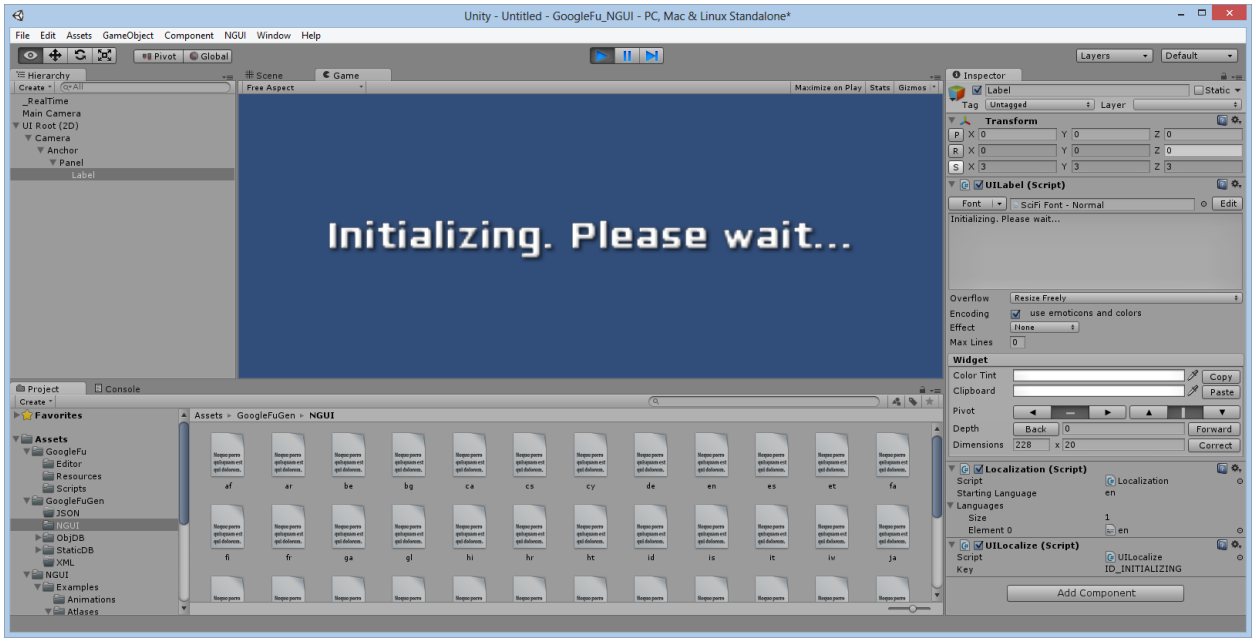



Each file is written in a format that NGUI expects for proper localization

```
1  Flag = Flag-en
2  ID_INITIALIZING = Initializing. Please wait...
3  ID_AUTHENTICATING = Authenticating with Google. Please wait...
4  ID_IMPORTING_MESSAGE = Importing Database. Please wait...
5  ID_SETTINGS = Settings
6  ID_WORKBOOKS = Workbooks
7  ID_CREDENTIALS = Credentials
8  ID_LANGUAGE = Language
9  ID_PATHS = Paths
10 ID_TOOLS = Tools
11 ID_HELP = Help
12 ID_HELP_MAIN = Main
13 ID_HELP_LOCAL = Localization
14 ID_HELP_DB = Database
15 ID_GENERATE_PATHS = Automatically Create Paths
```

Use the NGUI UILocalize script to designate a row ID for the  UILabel, and the NGUI Localization script to load the languages.

English:

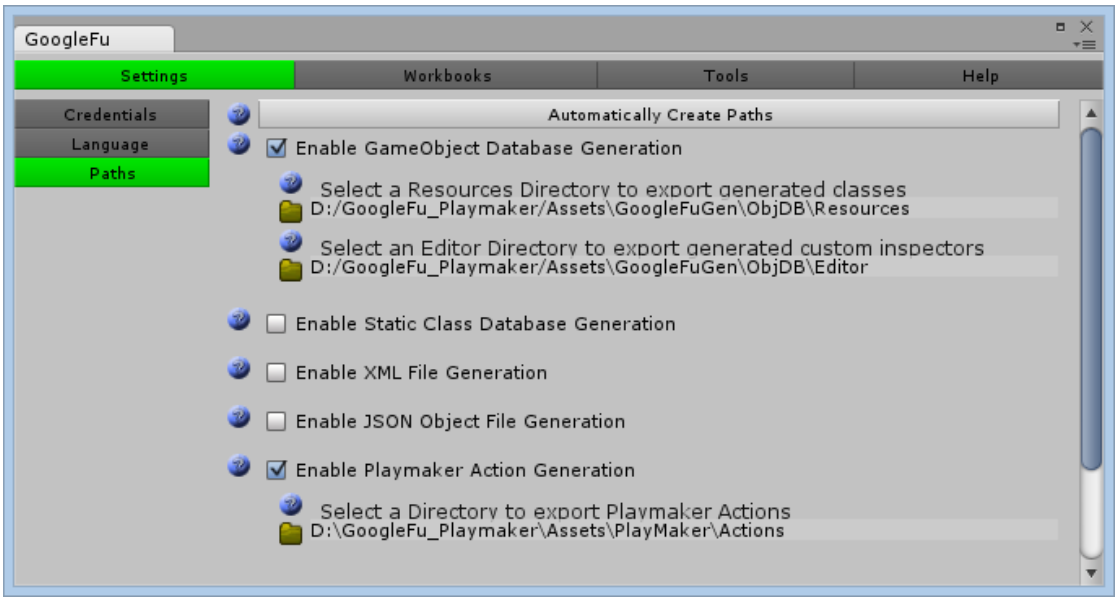



French:

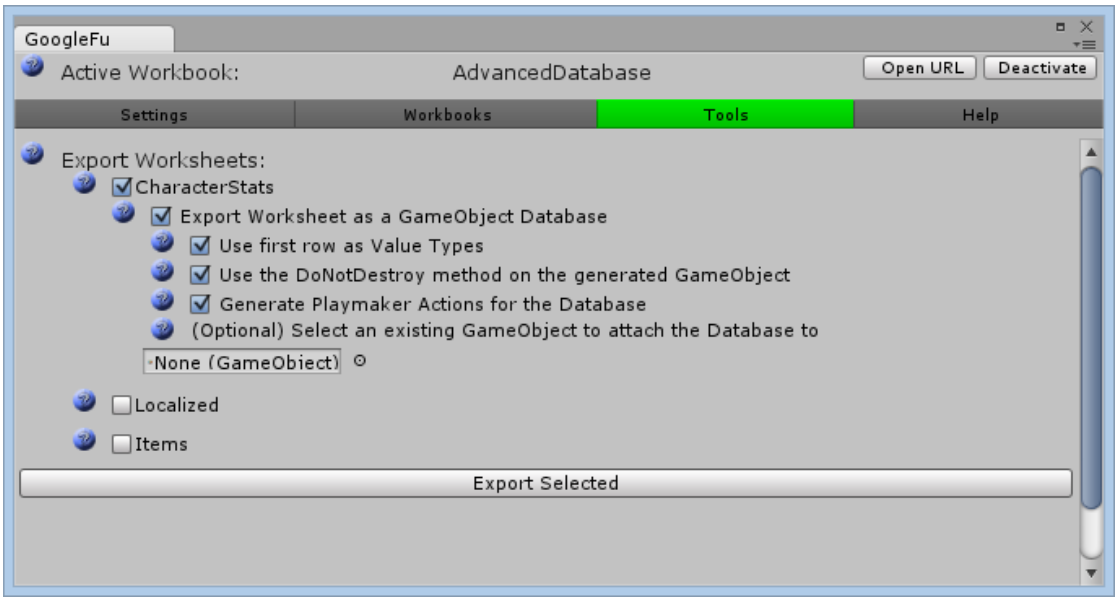
## Using GoogleFu with Playmaker

If Playmaker is present in the project, GoogleFu will recognize it* and allow you to generate Playmaker actions as part of the Object Database Export

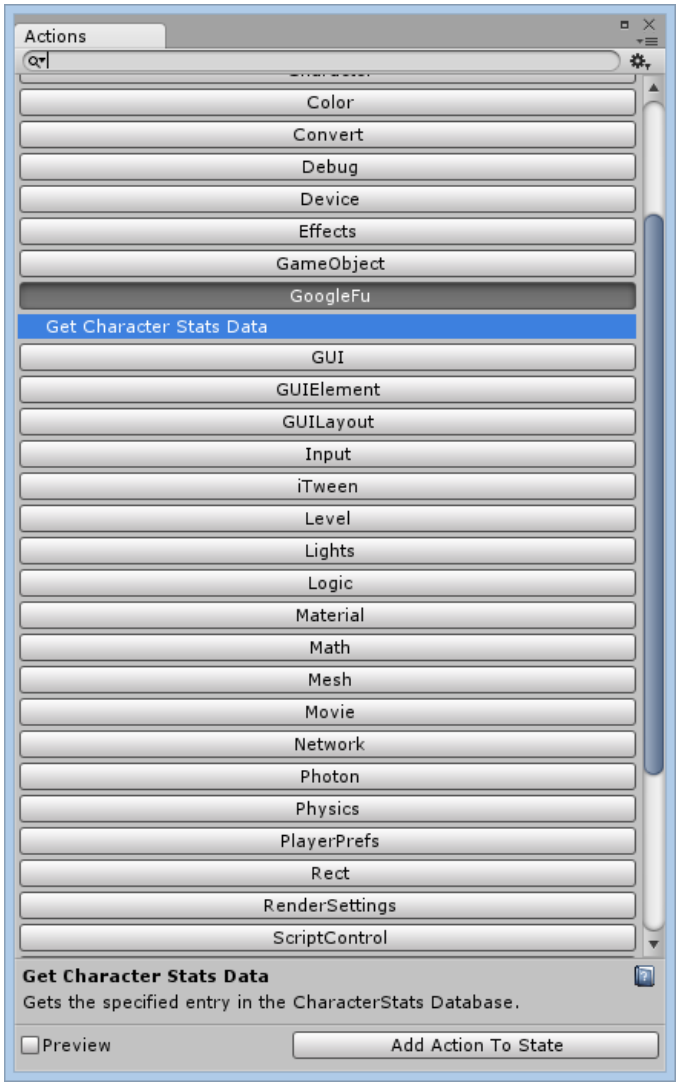
*GoogleFu looks for a specific file that Playmaker provides.



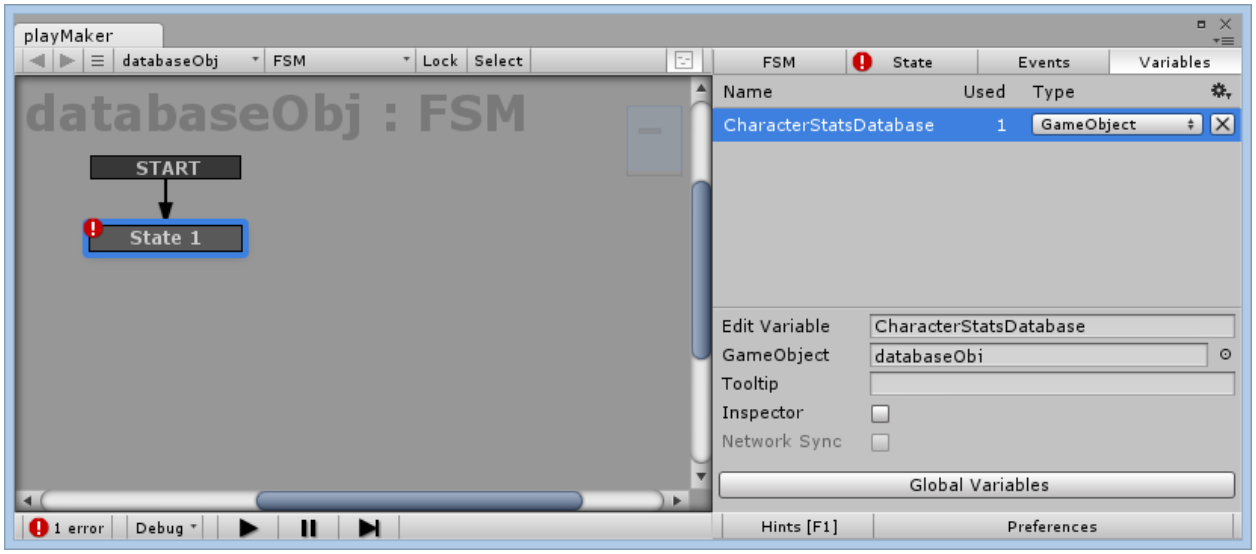

Once you have activated a workbook and selected a worksheet to export as a GameObject Database, you will be given the option to generate Playmaker Actions for the database.



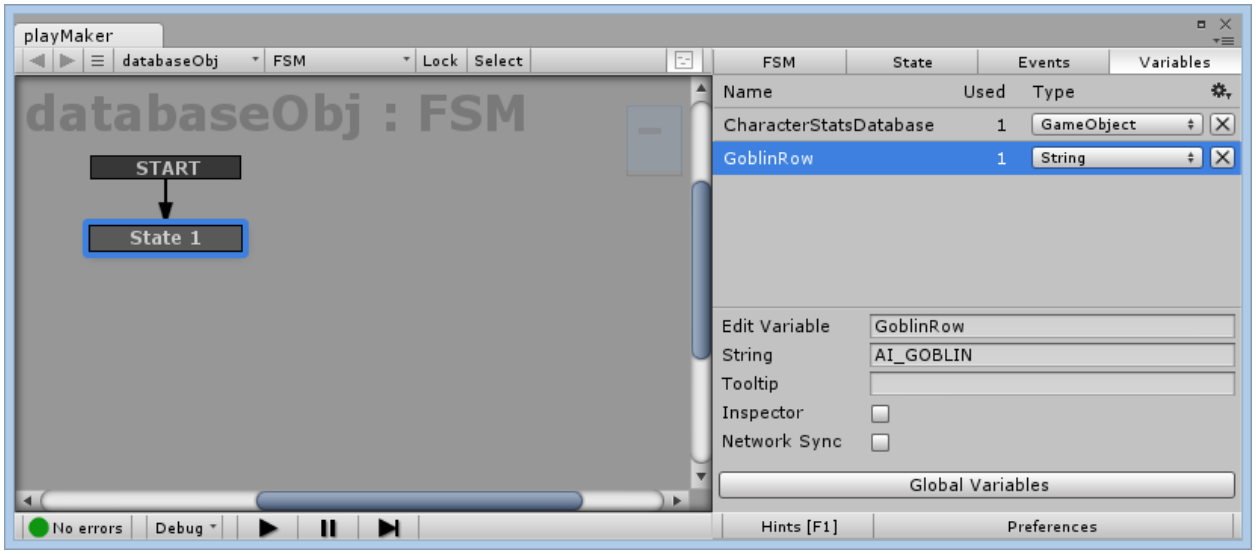
When you have generated the database, create a Playmaker FSM on a Game Object and open the Action Browser. A new option will be available for GoogleFu, and the Action that has been generated will be named after your worksheet



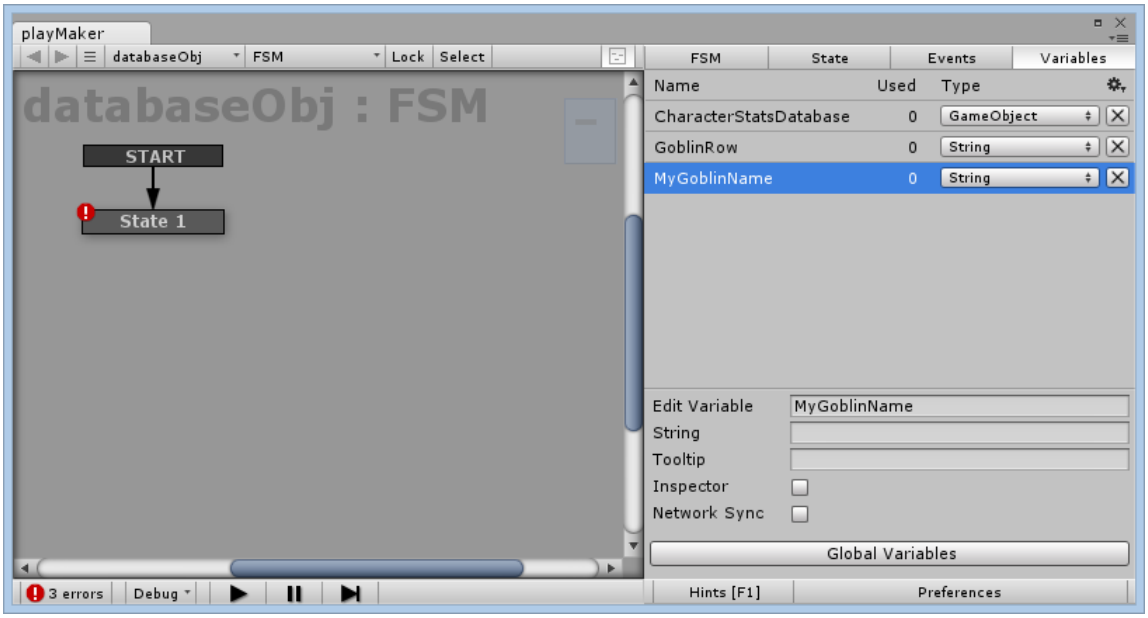
You will need to supply the object that you attached the database to via a Playmaker GameObject variable



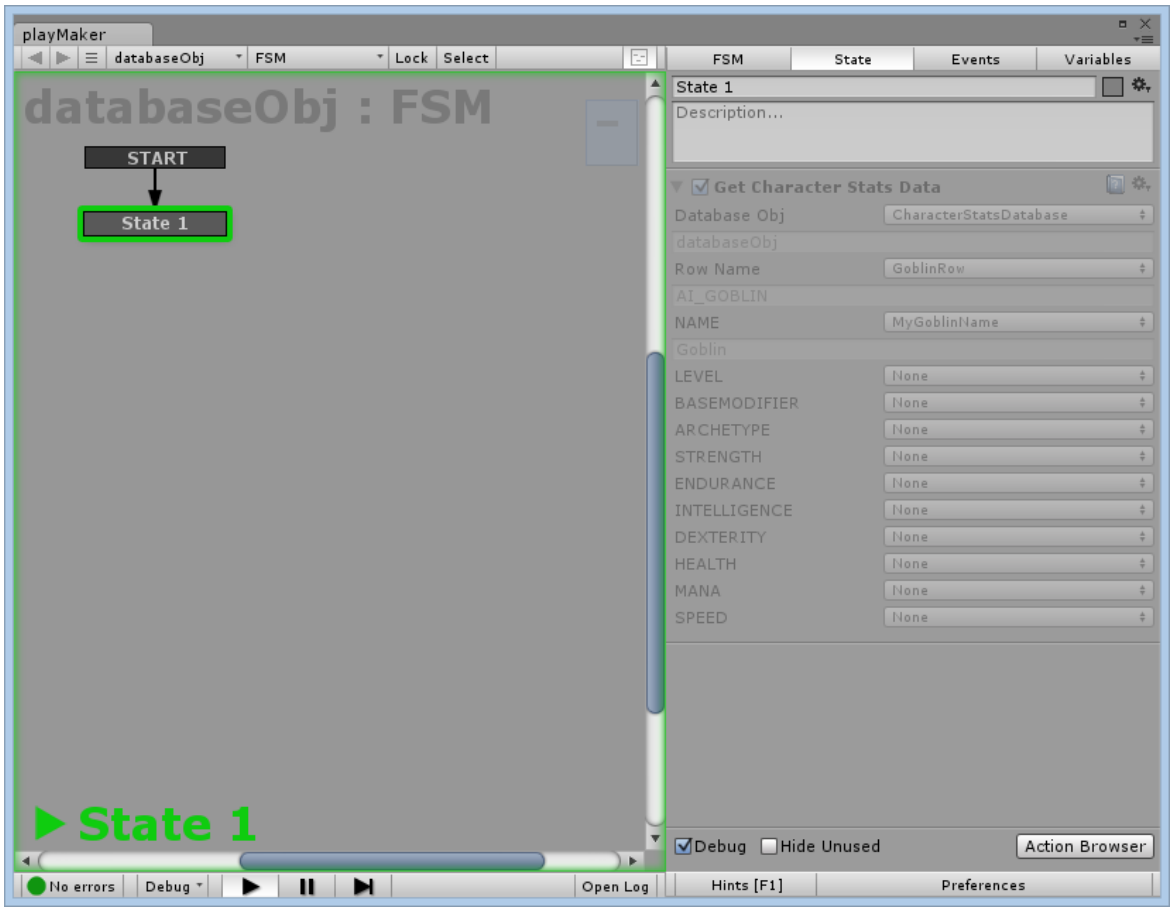

You will also need to supply the Row ID you want to access with a Playmaker String Variable



Any data you wish to retrieve from the database will also need to be stored in variables




Once you supply the State with the relevant information, all you need to do is Play, once the action is executed, all of the data will be pulled from the database and placed in the variables.



# Source Code, Third Party, and Copyright information

We want to be 100% transparent when it comes to GoogleFu accessing your personal Google Account, and allow you to rest at ease that there is nothing shady going on behind the scenes, so the full version of GoogleFu has not been compiled into a DLL.

Always, always keep a backup of your data. GoogleFu does overwrite existing files when it generates the database.

The DLL's that are included with GoogleFu are redistributed as part of the Google GData API. They have not been modified in any way, and full source code for the API is available through Google.



Tell your friends about GoogleFu. We are a couple of guys just trying to make a living, not some huge faceless corporation. Also, please rate us on the Unity Asset Store if you like GoogleFu. If not, let us know, and we'll try to fix the problem!

http://www.litteratus.net

# FAQ

## GoogleFu is saying that my login is invalid, but I KNOW my credentials are correct

This can happen in the case of having 2 step authentication turned on in your Google account. We recommend that you set up a Google account specifically for your project, and leave 2 step authentication turned off.

This also happens if your build target is the Web Player. Unity handles web requests differently between the build targets, and the Web Player target is restricted from Coss Domain traffic. Because GoogleFu relies on Unity networking to communicate with Google, it currently does not work. There are workarounds, but none of which are particularly elegant. We will continue to try to develop a good working solution for this issue, but for now all you can do is use the Standalone build target while using GoogleFu, and switch to the Web target after you have synced.

## Can I use GoogleFu to access my Google Spreadsheet at runtime?

No. GoogleFu is strictly a Pre-Build process. It simply streamlines pulling the information from a Google Spreadsheet and saving it in a way that is easily accessible from the game. Communication with Google can sometimes be very slow, and is not suited for real-time network traffic. Aside from that, it would also require you to place your Google Credentials directly into the code, which is a huge security risk. SQL databases that are accessed via PHP scripts on a webserver is the fastest, most secure way to handle runtime database access.

## I am getting a crash! What do I do?

Please contact us! We want to know exactly what you did when the crash happened. We are more than willing to help get you back up and running immediately, and will update GoogleFu with a fix as soon as possible!

contact@litteratus.net

Change Log

1.0.5
Moved DLL's into the Editor directory to prevent them from being included in final builds*
Added support for Diakon Forge localization file export

*If you have previously installed GoogleFu, updating this package will NOT remove the old DLL's from the project. You must manually remove all DLL's contained in the Assets/GoogleFu directory, or remove GoogleFu from the project entirely, and re-import. This will not remove any of your saved settings.

1.0.4
Fixed issues regarding Vector2, Vector3, and Quaternion types in the Custom Inspector code generation

1.0.3
Rearranged file structure to remove icons from being included into final builds
Removed a Using directive from a file that was causing build issues

1.0.2
Recompiled the NewtonsoftJson.dll. It should play nice with other plugins now

1.0.1
Initial version.